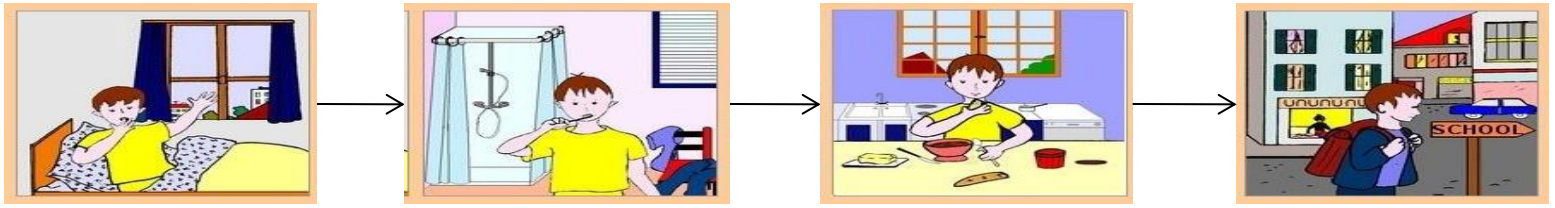
CSE101-Part-1(control structure)

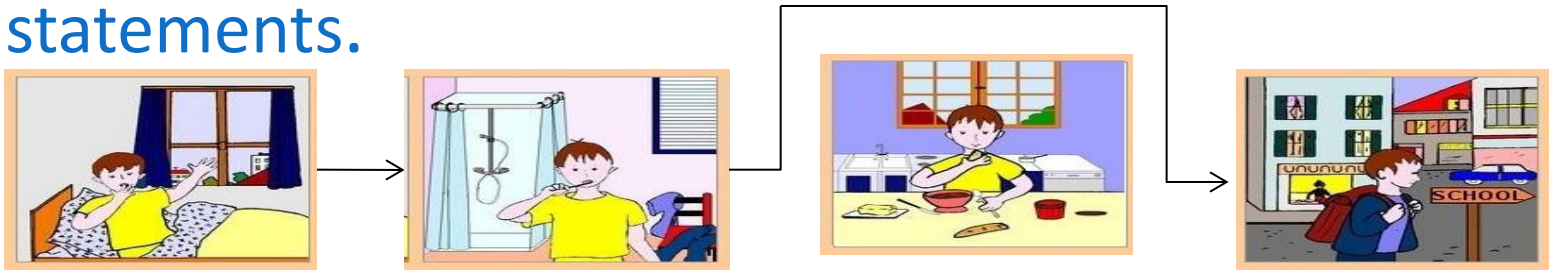
- Control structures(Decision control statements/ or Condition Statements)

Program

- Program is a set of instruction executed one by one.



- Depending upon the circumstances sometimes it is desirable to alter the sequence of execution of statements.



1. Wake up;
2. Get ready;
3. If you have enough time, then eat breakfast;
4. Go to school.

Control Statements

- The C language programs until now follows a sequential form of execution of statements.
- C language provides statements that can alter the flow of a sequence of instructions. These statements are called control statements.
- These statements help to jump from one part of the program to another. The control transfer may be conditional or unconditional.

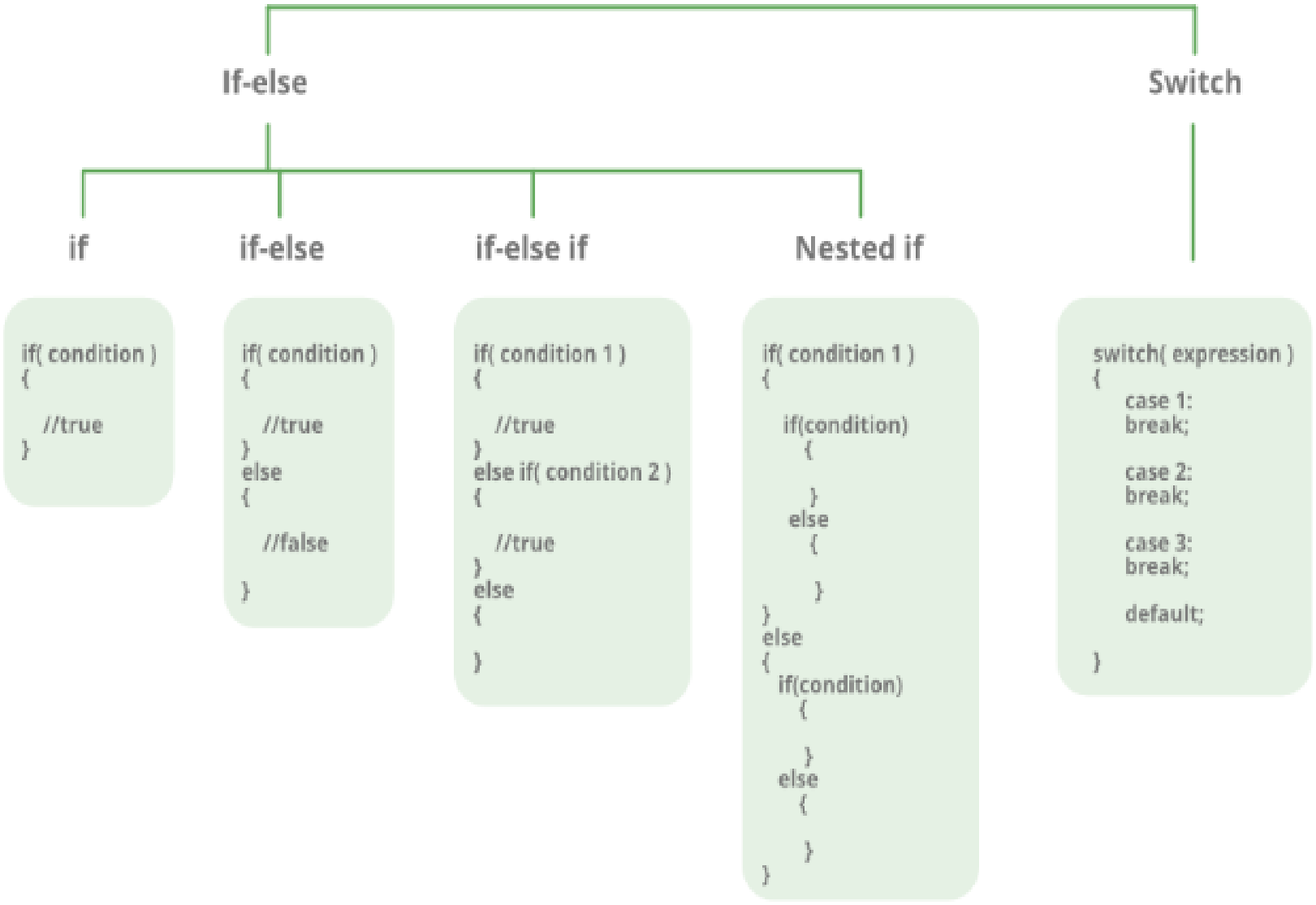
Control Structure

- A control structure refers to the way in which the programmer specifies the order of executing the statements.
- Three control structures
  - Sequence structure
    - Programs are executed sequentially by default.
  - Selection structures(Condition)
    - `if`, `if…else`, if-else-if, Nested-if ,`switch`
  - Repetition structures (iteration)
    - `while`, `do…while`, `for`

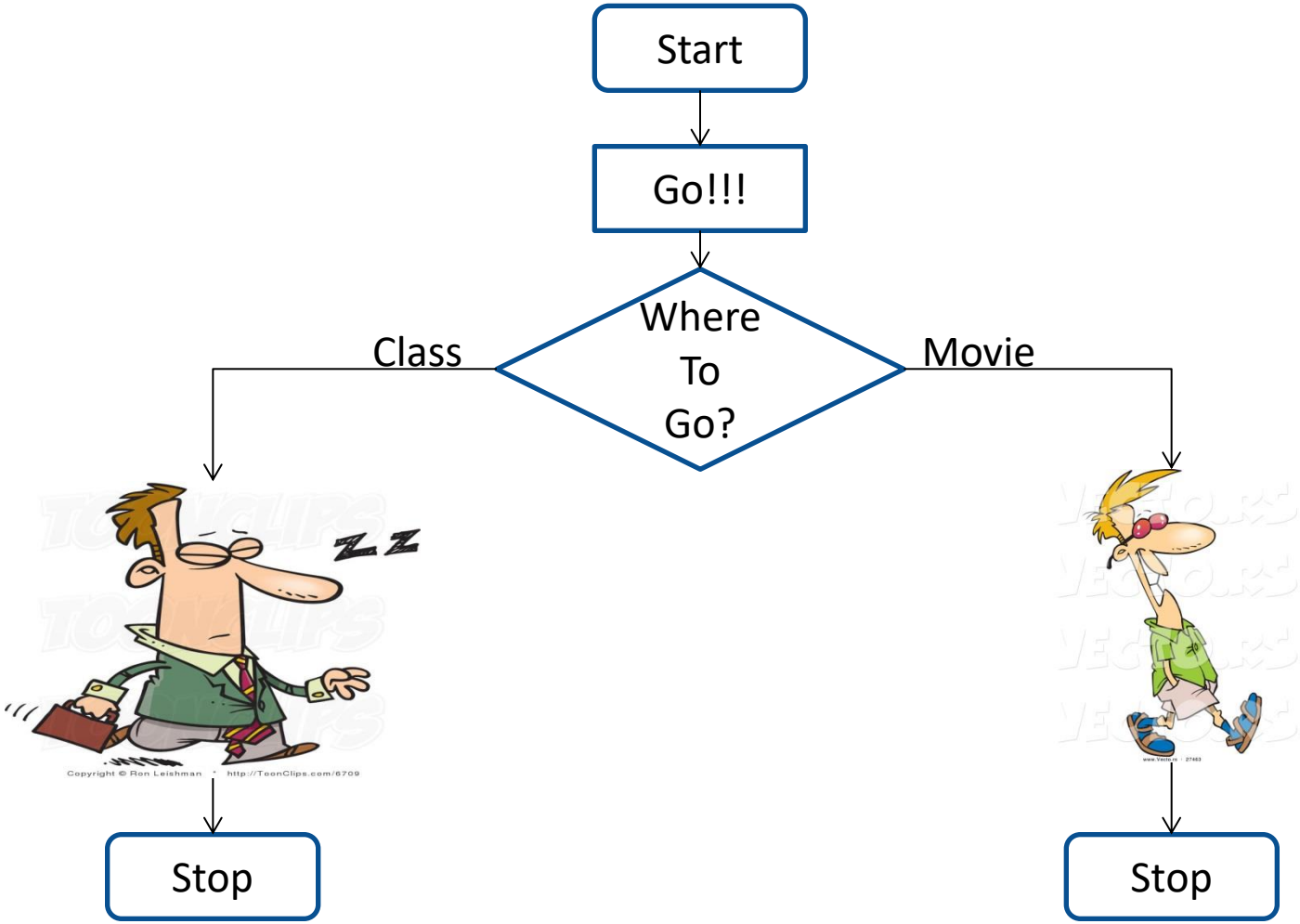
Condition Statements(or Decision control statements or Branching statements)

- The C condition statements or the decision statements, checks the given condition
- Based upon the state of the condition, a sub-block is executed.
- Decision statements are the:
  - *if statement*
  - *if-else statement*
  - *If-else-if statement*
  - *Nested if statement*
  - *switch statement*

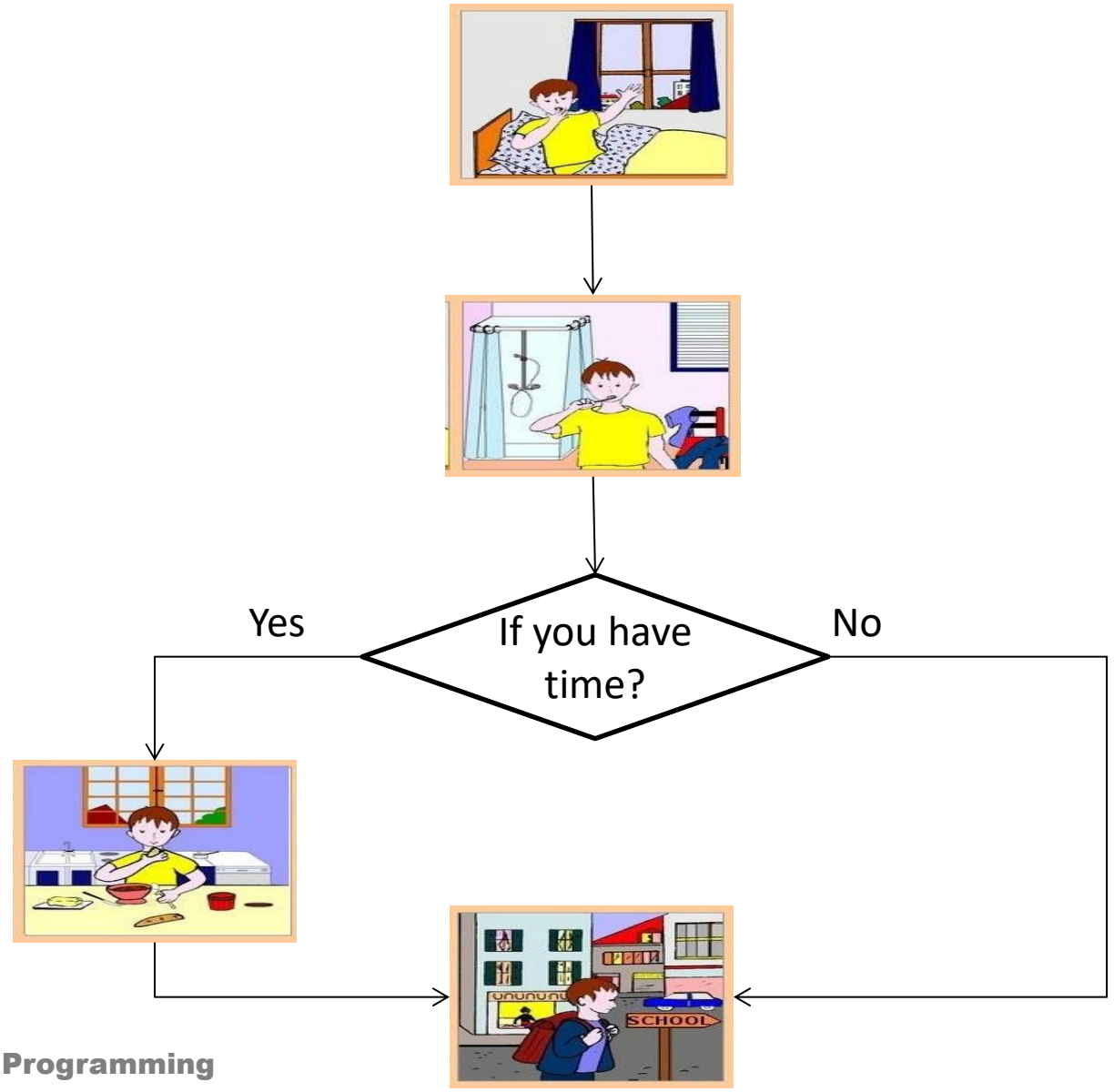
Decision Making



Daily routine



if statement



`if` Statement

- If statement
  - It is decision making statement uses keyword `if`.
  - It allows the computer to evaluate the expression first
    - and then, depending on whether the value is 'true' or 'false', i.e. non zero or zero it transfers the control to a particular statement.

A decision can be made on any expression.
zero - `false`
nonzero - `true`
Example:
`3 < 4` is `true`

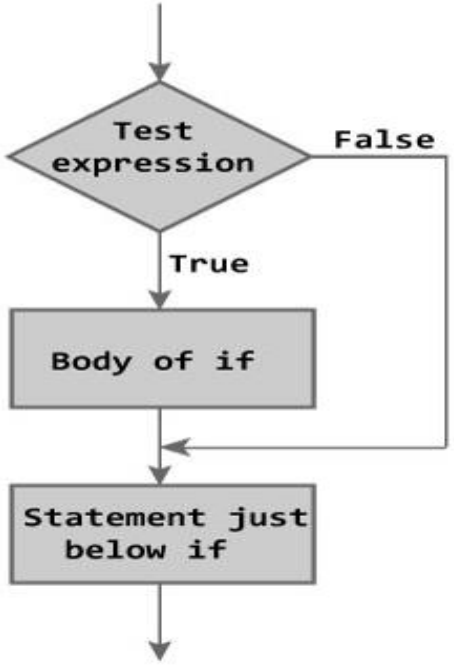
if Statement

Syntax

```
if (expression)
statement;

or

if (expression)
{
  block of statements;
}
```



`if` Statement

- The *if statement* has the following syntax:

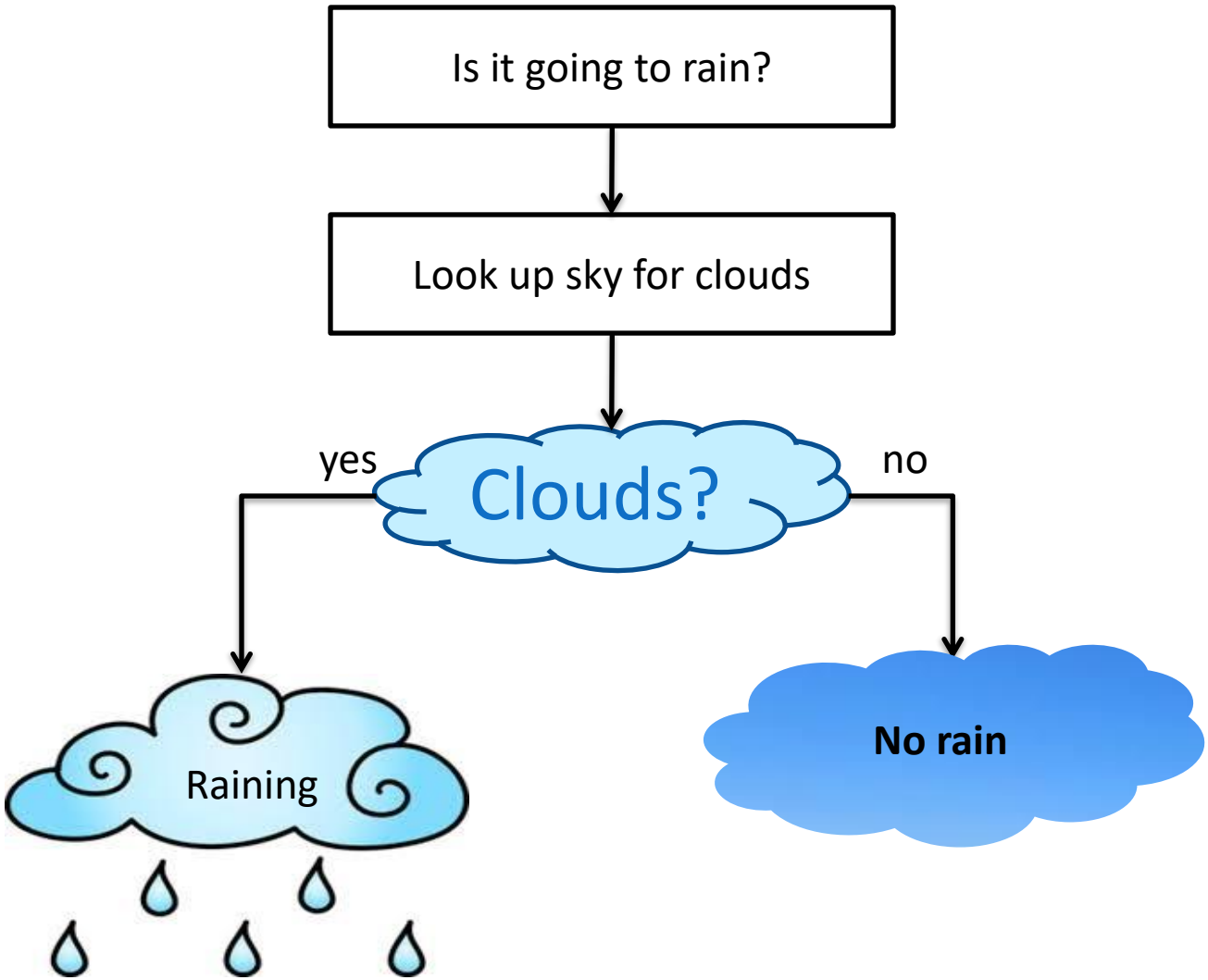
`if` is a C reserved word

The *`condition`* must be a boolean expression. It must Evaluate to either non-zero or zero.

```
if ( condition )/* no semi-colon */
    statement;
```

If the *`condition`* is non-zero, the *`statement`* is executed. If it is zero, the *`statement`* is skipped.

Rain ???

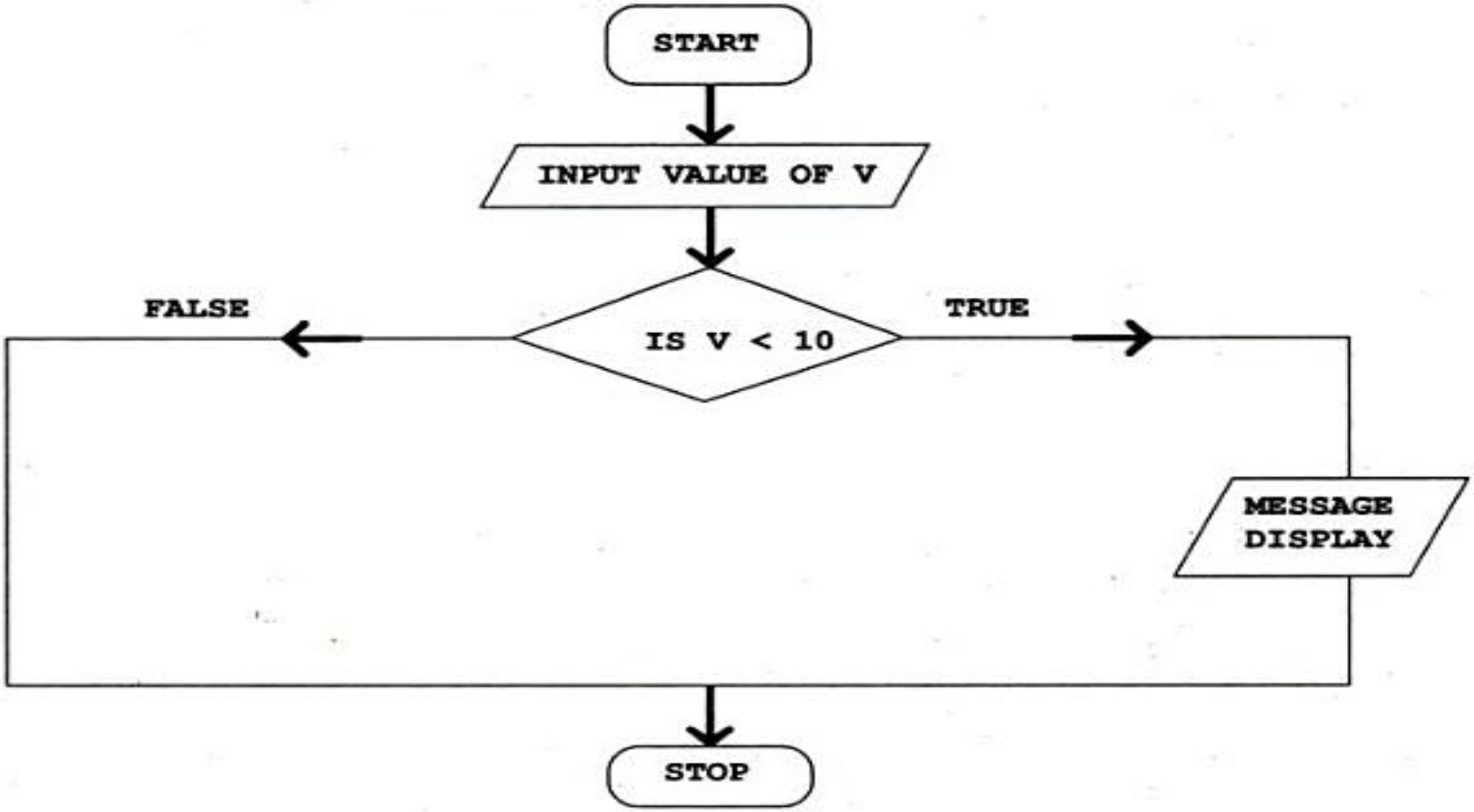

Is it going to rain?
Look up sky for clouds
Clouds?
yes
no
Raining
No rain

```
#include<stdio.h>
int main()
{
 int v;
 printf("Enter the number :");
 scanf("%d", &v);
 if(v<10)
   printf("number is less  than 10");
return 0;
}
```

Program to check whether number is less than 10.

```
Enter the number: 6
Number is less than 10
```

Control Flow



`if..else` statement

If you have time?

Yes

No

Grab something to eat along

`if..else` statement

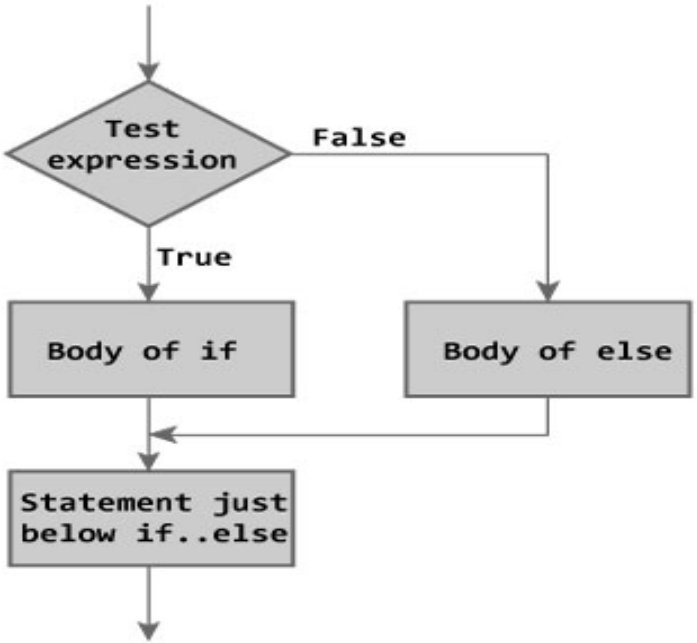
- The `if` statement executes only when the condition following if is true.
- It does nothing when the condition is false.
- The `if..else` statement takes care of the true and false conditions.

`if..else` statement

- `if..else` has two blocks.
- One block is for `if` and it is executed when condition is **non-zero**(true).
- The other block is of `else` and its executed when condition is **zero** (false).

Syntax

```
if (expression)
{
  block of statements;
}
else
{
  block of statements;
}
```



`if..else` statement

- The `else` statement cannot be used without `if`.
- No multiple `else` statements are allowed with one `if`.
- `else` statement has no expression.
- Number of `else` cannot be greater than number of `if`.

```
#include<stdio.h>
int main()
{
 int a;
 printf("Enter the number :");
 scanf("%d", &v);
 if(v<10)
   printf("number is less than 10");
 else
   printf("number is greater than 10");
return 0;
}
```

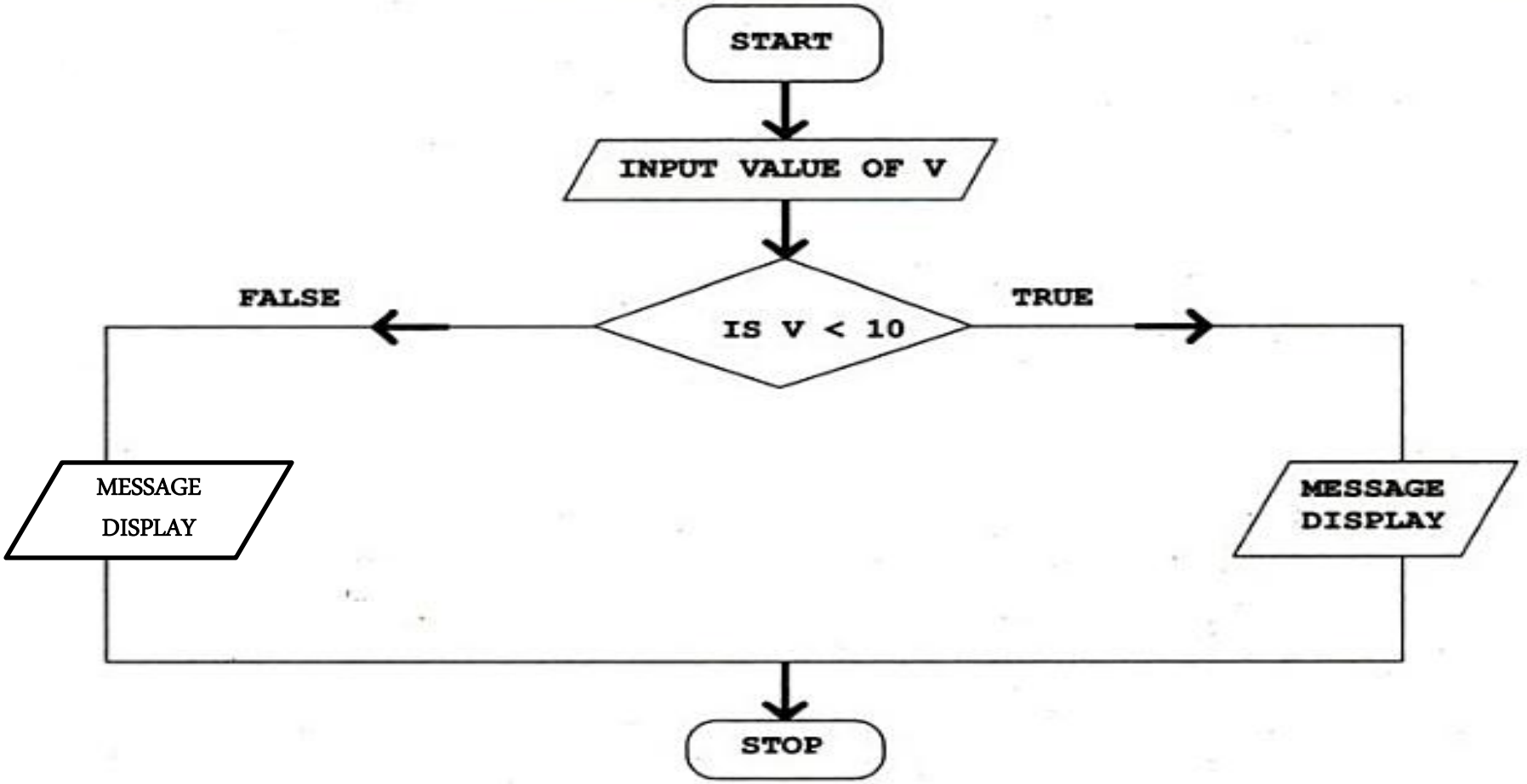
Example : Program to check whether number is less than 10.

```
Enter the number: 7
Number is less than 10
```

or

```
Enter the number: 100
Number is greater than 10
```

Control Flow



Q1

What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 5;
    if (x < 1)
        printf("hello");
    if (x == 5)
        printf("hi");
    else
        printf("no");
    return 0;
}
```

A. hi

B. hello

C. no

D. error

Q2

What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 0;
    if (x == 0)
        printf("hi");
    else
        printf("how are u");
        printf("hello");
    return 0;
}
```

A. hi

B. how are you

C. hello

D. hihello

Q3

What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int x = 5;
    if (x < 1);
        printf("Hello");

}
```

A. Nothing will be printed

B. Compile time error

C. Hello

D. Logical error

Q4

```
#include<stdio.h>
int main()
{
float x=2.3;
if(x==2.3)
{
printf("Hi");
}
else
{
printf("Hello");
}
return 0;
}
```

A. Hi
B. Hello
C. Compile time error
D. None of these

Q5

```
#include<stdio.h>
int main()
{
int x=-1;
if(x)
{
printf("Hi");
}
else
{
printf("Hello");
}
return 0;
}
```

A. Hi
B. Hello
C. Compile time error
D. None of these

Q6

What is the output of this C code?

```
#include <stdio.h>
int main()
{
    float f = 0.1;
    if (f == 0.1)
        printf("True");
    else
        printf("False");
    return 0;
}
```

A. True

B. False

C. Compile time error

D. None of these